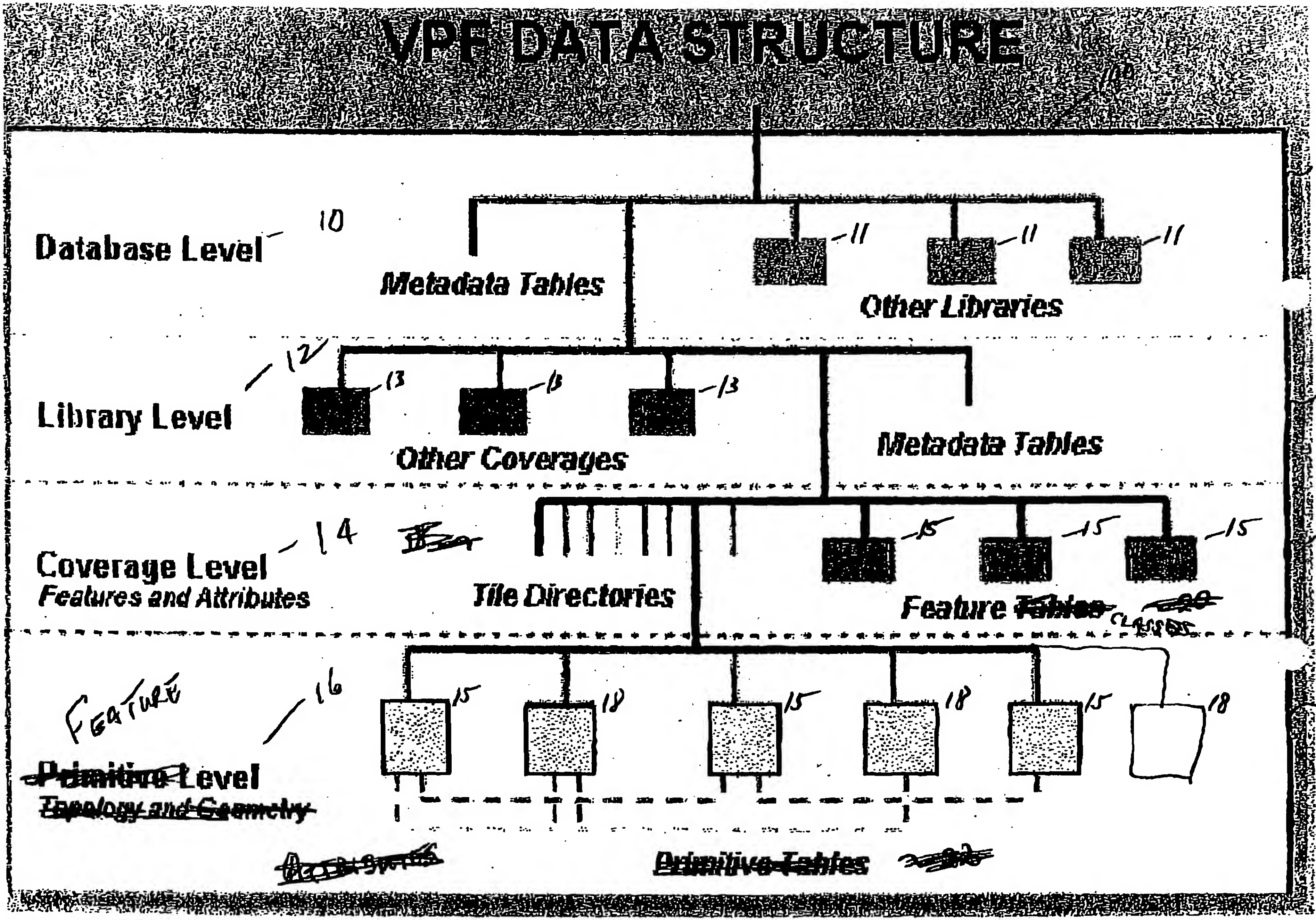
VPF DATA STRUCTURE



Database Level

Metadata Tables

Other Libraries

Library Level

Other Coverages

Metadata Tables

Coverage Level
Features and Attributes

Tile Directories

Feature ~~Tables~~ CLASSES

FEATURE ~~Primitive~~ Level
~~Topology and Geometry~~

~~Primitive Tables~~

Prior Art Fig 1

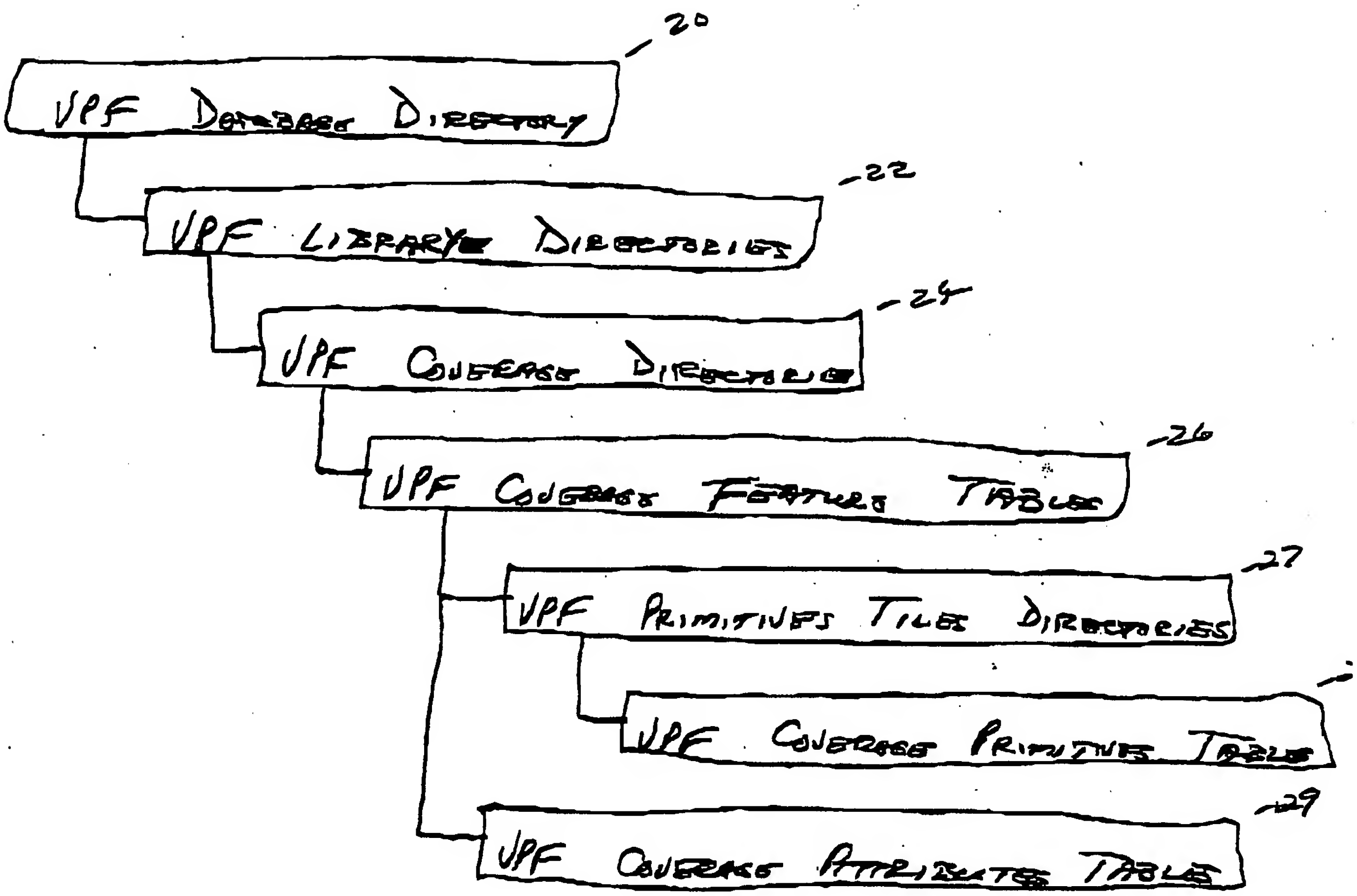

20
VPF Database Directory
22
VPF Librarye Directories
24
VPF Coverage Directories
26
VPF Coverage Feature Table
27
VPF Primitives Tiles Directories
VPF Coverage Primitive Table
29
VPF Coverage Attributes Table

Fig 2

Prior Art

~~left and right faces. Left and right fa~~

~~th these definitions, winged-edge topol~~

~~entation of neighboring nodes, edges and~~

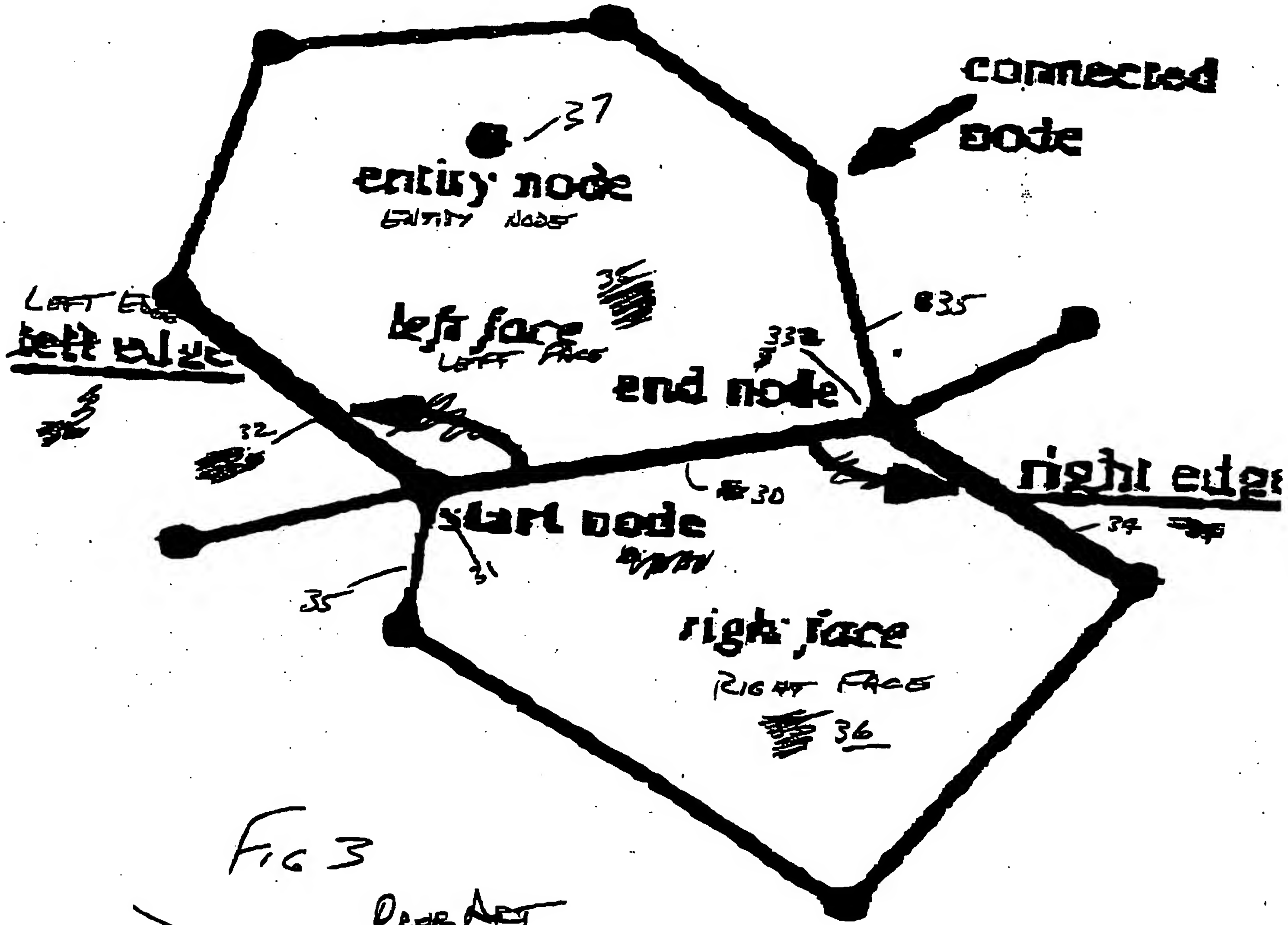


Fig 3

Prior Art

~~Figure 1. Winged-Edge Topology~~

~~provide a mutually exclusive and colle~~

~~h partition a geographic region. In ot~~

~~efined by a set of faces and edges that do~~

~~Stages of Initialization~~

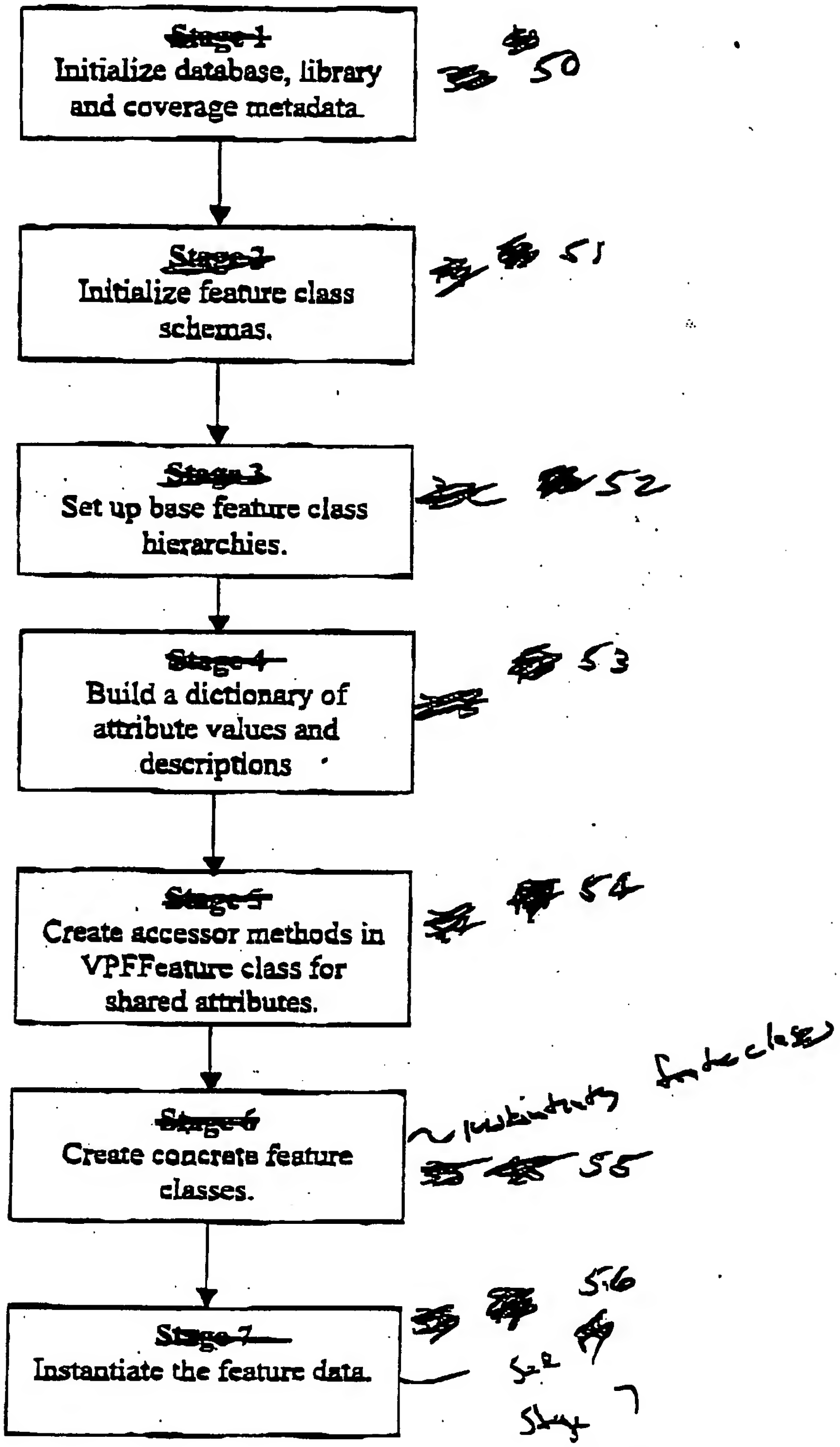

~~Stage 1~~
Initialize database, library and coverage metadata.
~~33~~ ~~50~~ 50
~~Stage 2~~
Initialize feature class schemas.
~~34~~ ~~51~~ 51
~~Stage 3~~
Set up base feature class hierarchies.
~~35~~ ~~52~~ 52
~~Stage 4~~
Build a dictionary of attribute values and descriptions
~~36~~ ~~53~~ 53
~~Stage 5~~
Create accessor methods in VPFFeature class for shared attributes.
~~37~~ ~~54~~ 54
~~Stage 6~~
Create concrete feature classes.
~ instantiate feature classes
~~38~~ ~~55~~ 55
~~Stage 7~~
Instantiate the feature data.
~~39~~ ~~56~~ 56
see ~~4~~
Stage 7
blowup
FIG 4

FIG ~~3~~ ~~4~~ 5

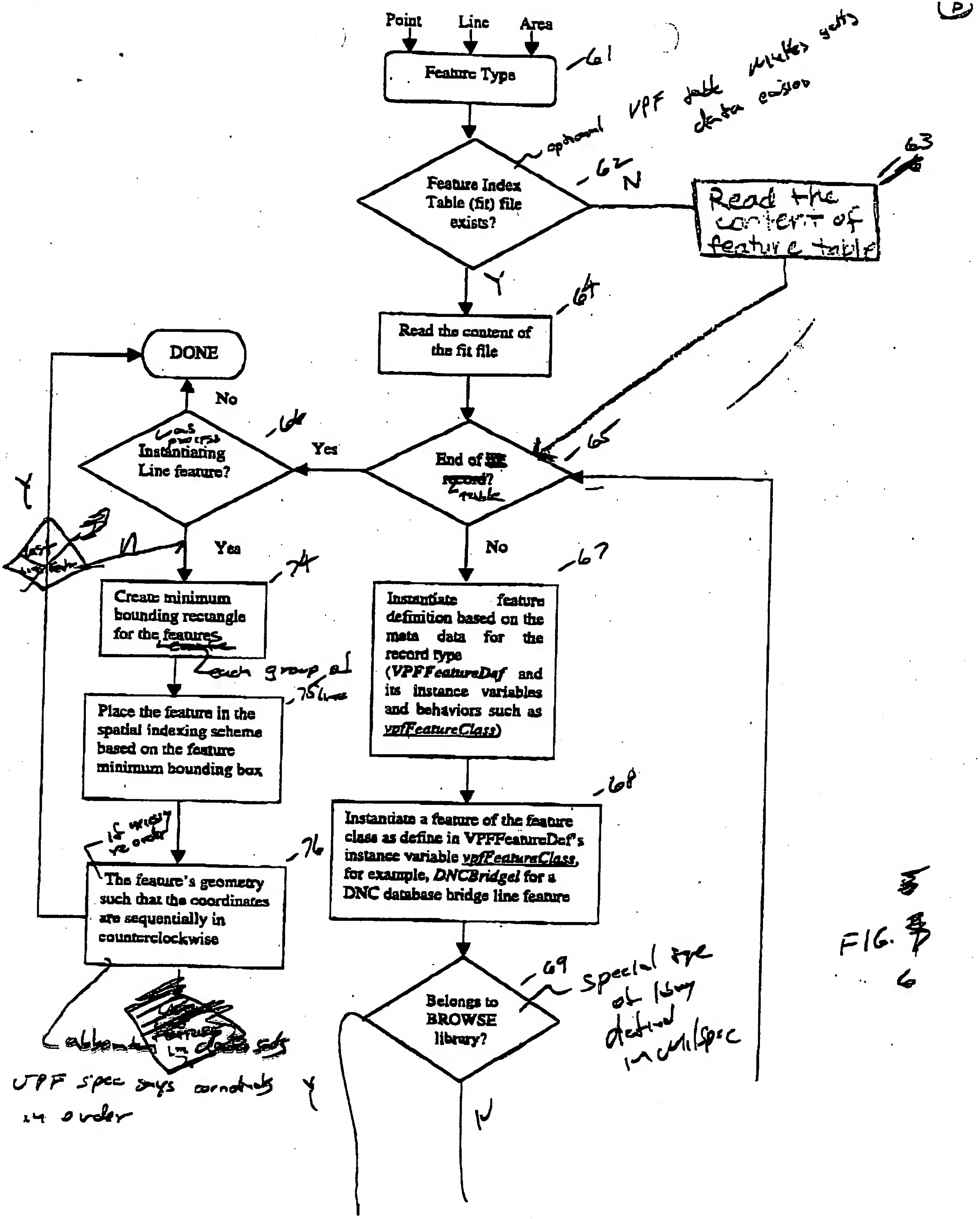
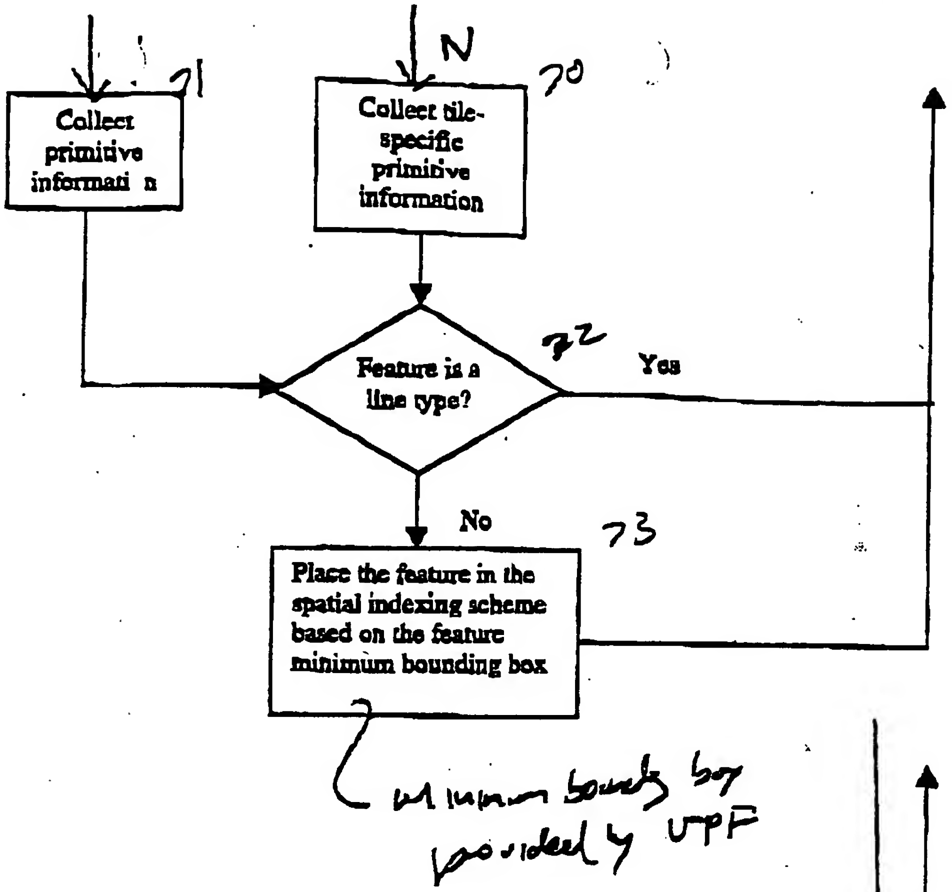

Point
Line
Area
Feature Type
61
optional VPF table makes gets data easier
Feature Index Table (fit) file exists?
62
N
63
Read the content of feature table
Y
Read the content of the fit file
64
DONE
No
Was processed Instantiating Line feature?
66
Yes
End of ~~fit~~ ~~record~~? table
65
Y
test list feature
N
Yes
No
Create minimum bounding rectangle for the features
each group of
74
75 lines
Place the feature in the spatial indexing scheme based on the feature minimum bounding box
67
Instantiate feature definition based on the meta data for the record type (VPFFeatureDef and its instance variables and behaviors such as vpfFeatureClass)
if not in order
76
The feature's geometry such that the coordinates are sequentially in counterclockwise
68
Instantiate a feature of the feature class as define in VPFFeatureDef's instance variable vpfFeatureClass, for example, DNCBridgeI for a DNC database bridge line feature
alternate in class sets
VPF spec says coordinates in order
Y
69
special type of library defined in milspec
Belongs to BROWSE library?
N

FIG. 6


N
71
70
Collect primitive informati n
Collect tile-specific primitive information
72
Feature is a line type?
Yes
No
73
Place the feature in the spatial indexing scheme based on the feature minimum bounding box
minimum bounding box provided by UPF

ckET 78,350

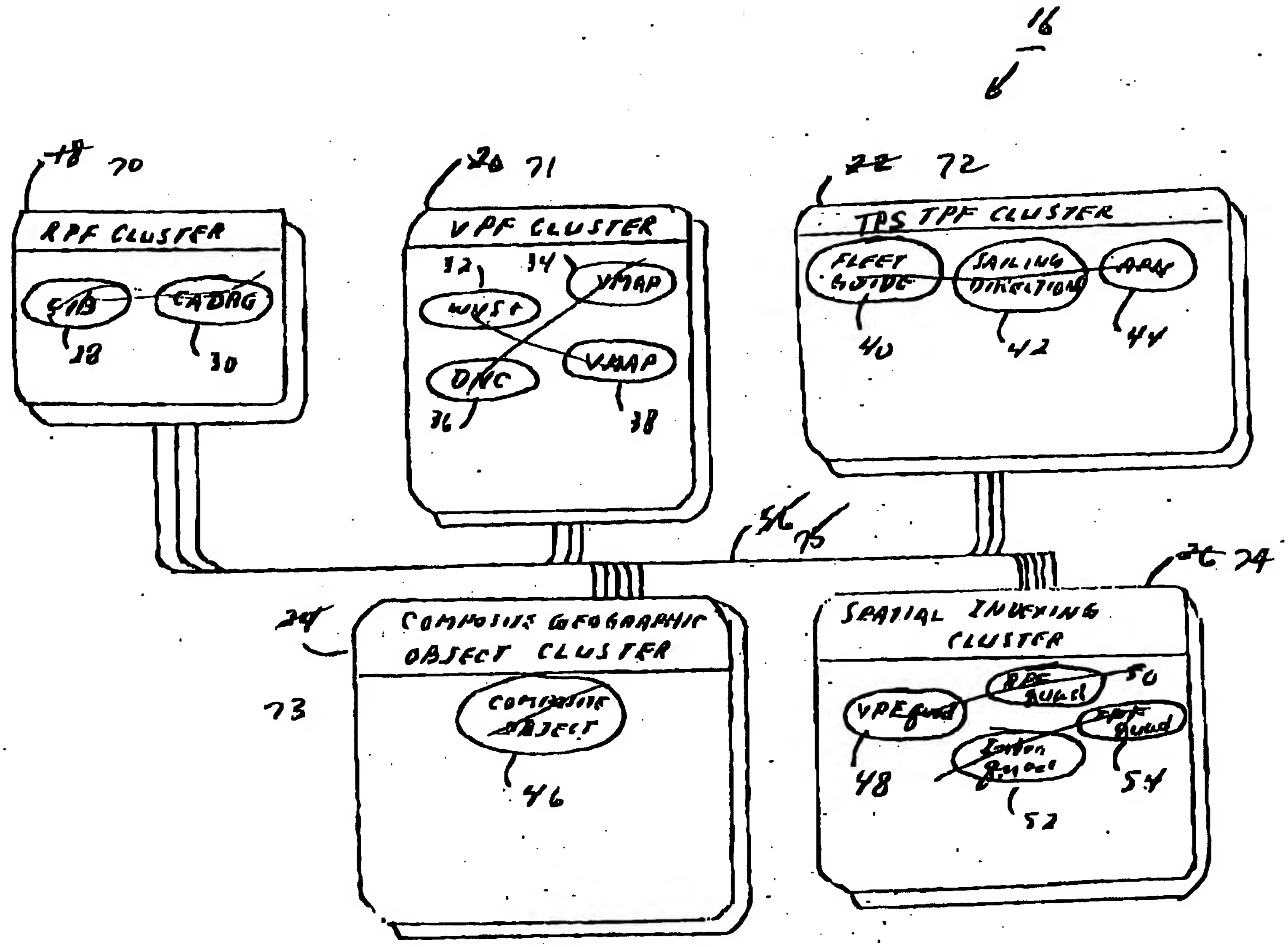


FIG 7

DOCKET 78,350

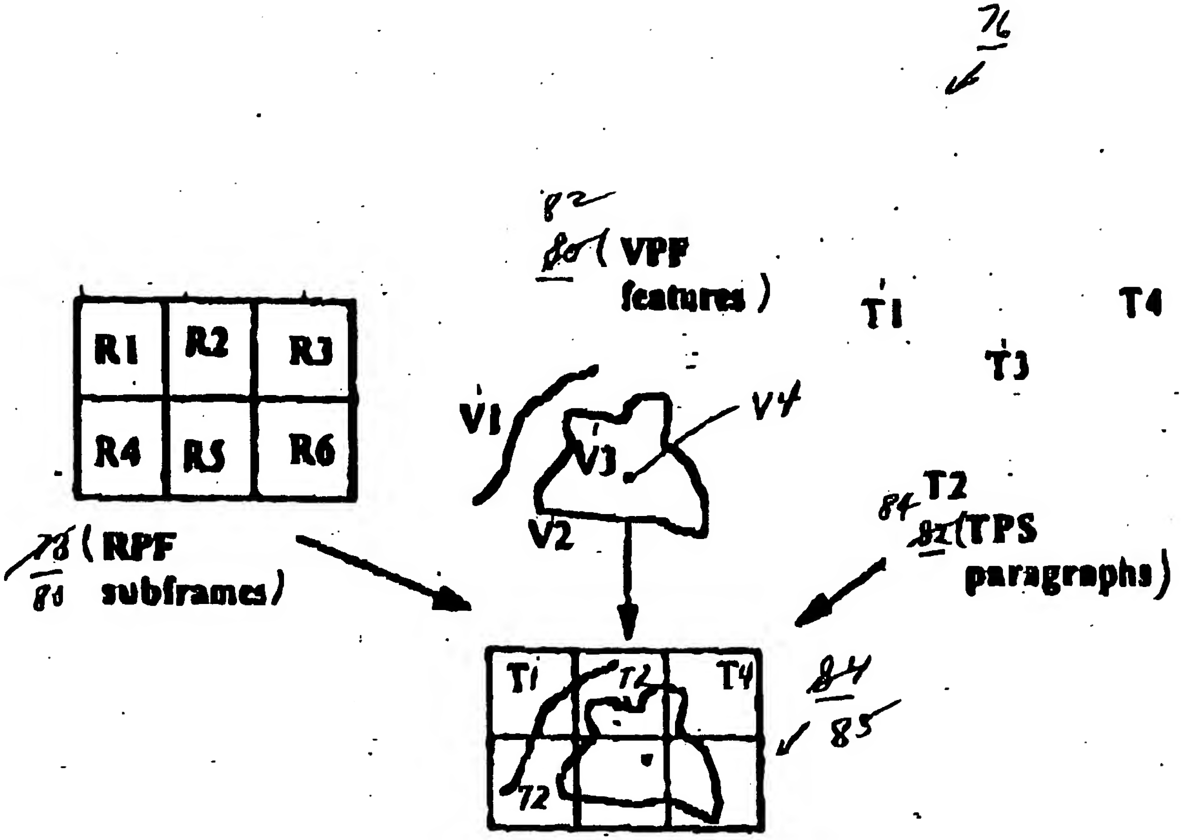


FIG. ~~X~~ 8

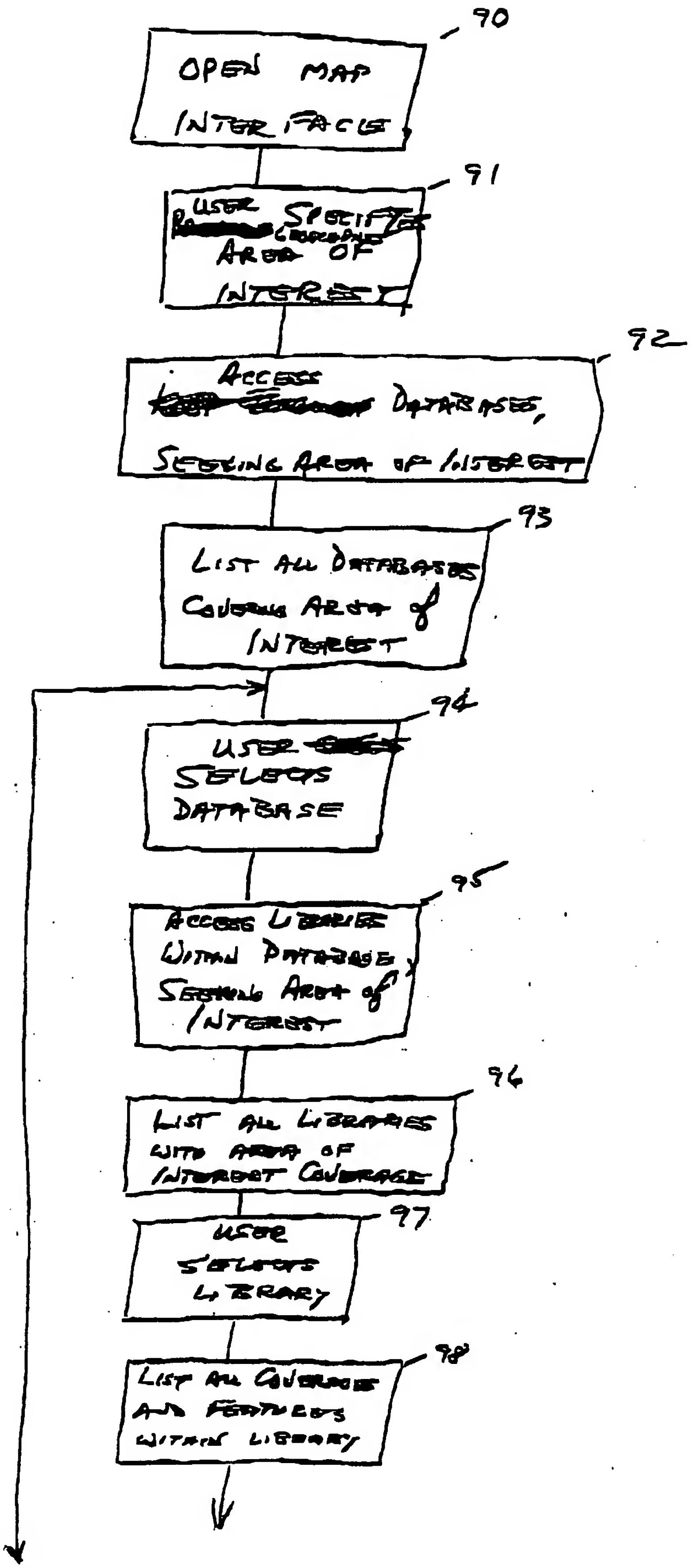

90
OPEN MAP INTERFACE
91
USER SPECIFIES GEOGRAPHIC AREA OF INTEREST
92
ACCESS DATABASES, SEEING AREA OF INTEREST
93
LIST ALL DATABASES COVERING AREA OF INTEREST
94
USER SELECTS DATABASE
95
ACCESS LIBRARIES WITHIN DATABASE, SEEING AREA OF INTEREST
96
LIST ALL LIBRARIES WITH AREA OF INTEREST COVERAGE
97
USER SELECTS LIBRARY
98
LIST ALL COVERAGES AND FEATURES WITHIN LIBRARY

FIG. 9

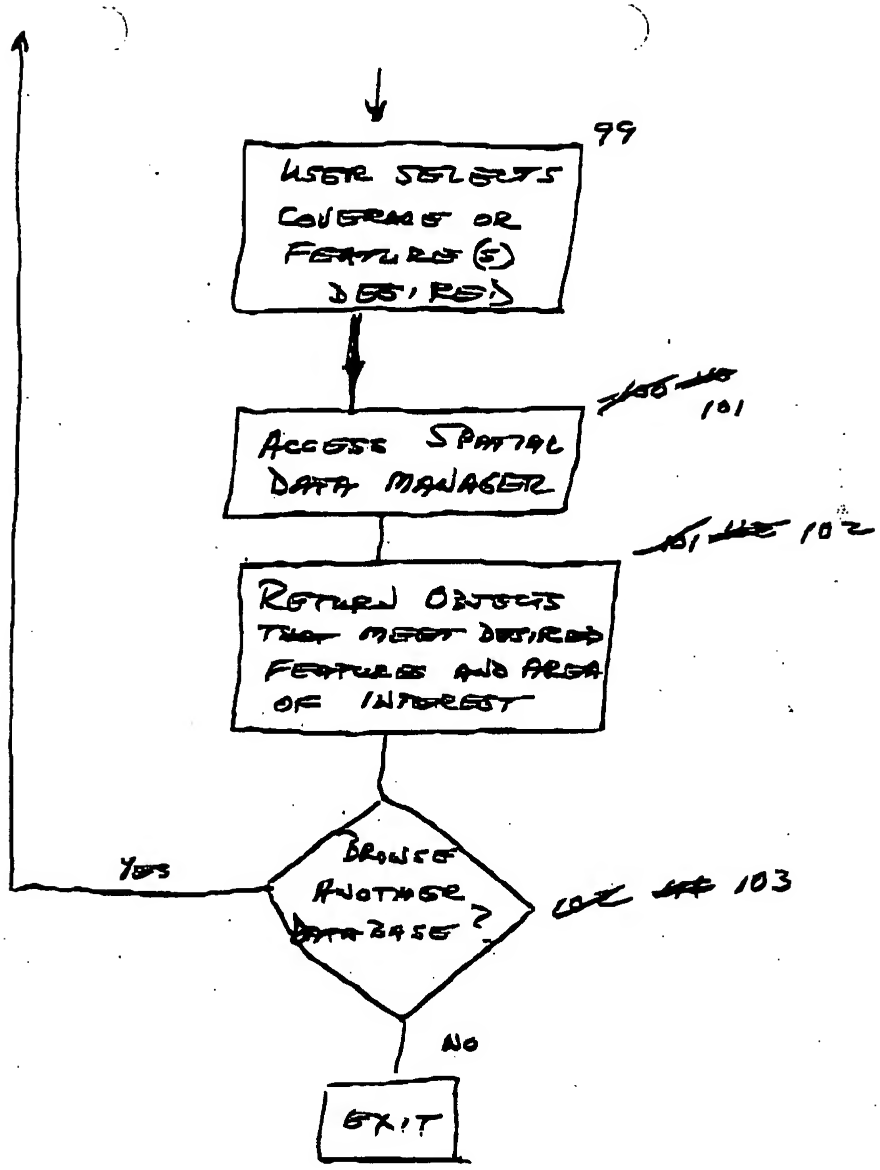
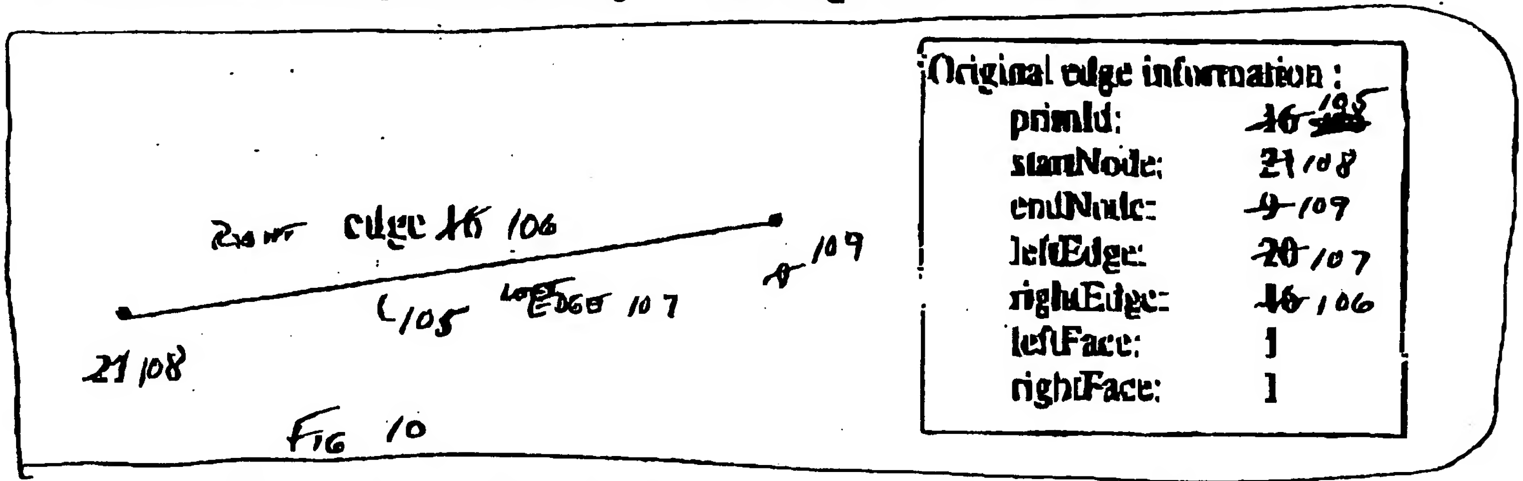

99
USER SELECTS COVERAGE OR FEATURE(S) DESIRED
101
ACCESS SPATIAL DATA MANAGER
102
RETURN OBJECTS THAT MEET DESIRED FEATURES AND AREA OF INTEREST
BROWSE ANOTHER DATABASE?
103
YES
NO
EXIT

FIG. 9

share *edge*'s startNode

~~Such an algorithm to determine the neighboring edges can be error-prone. To prevent this possibility, a graphical user interface (GUI) has been developed to allow the user to select neighboring edges. All incident edges to the edge in consideration are displayed. With buttons for **next**, **previous** and **select** actions, the user has the flexibility to view and select among all possible proximate edges.~~

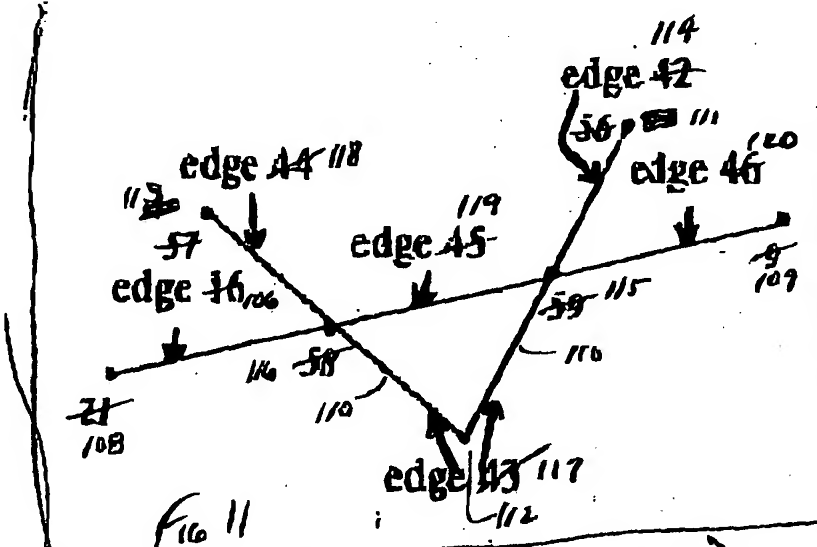
An example of an edge intersecting another edge is described below.



Original edge information :	
primId:	~~16~~ ~~105~~ 105
startNode:	~~21~~ 108
endNode:	~~9~~ 109
leftEdge:	~~20~~ 107
rightEdge:	~~16~~ 106
leftFace:	1
rightFace:	1

Figure 4. EdgePrim Before a New Line Feature is Added

~~Add a new line feature defined by three coordinate points at each vertex. Originally, this new feature is created as edge 42 with startNode and endNode of 56 and 57. The two lines intersect at two locations. Topology update should result in creation of two other edges for the new line feature and also for the intersecting line feature. The resulting winged-edge topology after adding a feature can be seen in Figure 5.~~



primId:	105 ~~16~~	~~45~~	~~46~~	~~44~~	~~43~~	~~42~~
startNode:	108 ~~21~~	~~58~~	~~59~~	~~58~~	~~59~~	~~56~~
endNode:	116 ~~58~~	~~59~~	~~9~~	~~57~~	~~58~~	~~39~~
leftEdge:	107 ~~20~~	~~44~~	~~42~~	~~16~~	~~46~~	~~42~~
rightEdge:	117 ~~45~~	~~45~~	~~16~~	~~44~~	~~45~~	~~45~~
leftFace:	1	1	1	1	1	1
rightFace:	1	1	1	1	1	1

Figure 5. EdgePrims After a New Line Feature is Added

Each feature is completely defined by its primitive information and maintains its own primitive information. Since each feature's *symbol* has direct object pointers to its *graphicElements* collection of spatial primitives, updating and maintaining topology is greatly facilitated.

CONCLUSIONS AND FUTURE WORK

We have demonstrated that VPF winged-edge topology can be updated in a reasonable manner within an object-oriented framework. We have observed that

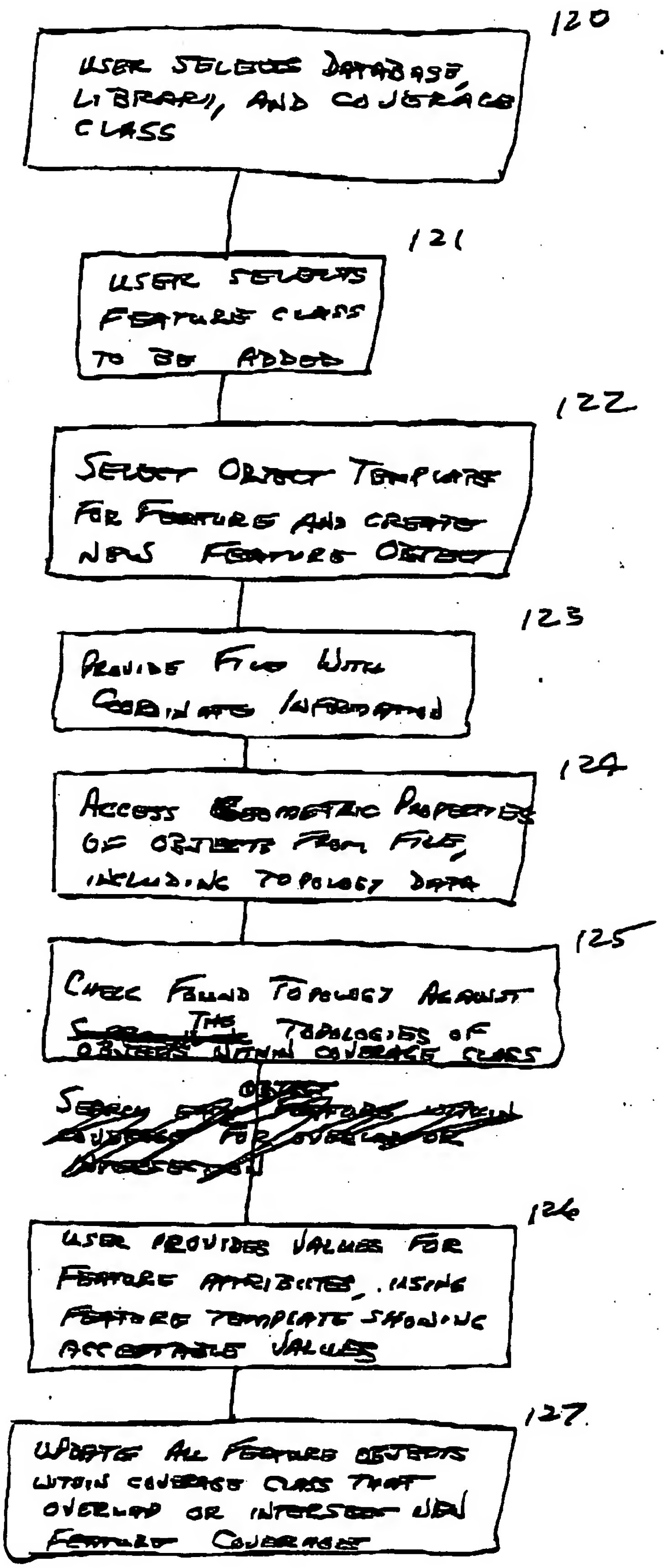

120
USER SELECTS DATABASE, LIBRARY, AND COVERAGE CLASS
121
USER SELECTS FEATURE CLASS TO BE ADDED
122
SELECT OBJECT TEMPLATE FOR FEATURE AND CREATE NEW FEATURE OBJECT
123
PROVIDE FILE WITH COORDINATE INFORMATION
124
ACCESS GEOMETRIC PROPERTIES OF OBJECTS FROM FILE, INCLUDING TOPOLOGY DATA
125
CHECK FOUND TOPOLOGY AGAINST ~~[illegible]~~ THE TOPOLOGIES OF OBJECTS WITHIN COVERAGE CLASS
~~SEARCH FOR OBJECT FEATURES WITHIN COVERAGE FOR OVERLAP OR INTERSECTION~~
126
USER PROVIDES VALUES FOR FEATURE ATTRIBUTES, USING FEATURE TEMPLATE SHOWING ACCEPTABLE VALUES
127
UPDATE ALL FEATURE OBJECTS WITHIN COVERAGE CLASS THAT OVERLAP OR INTERSECT NEW FEATURE COVERAGE

Fig 12

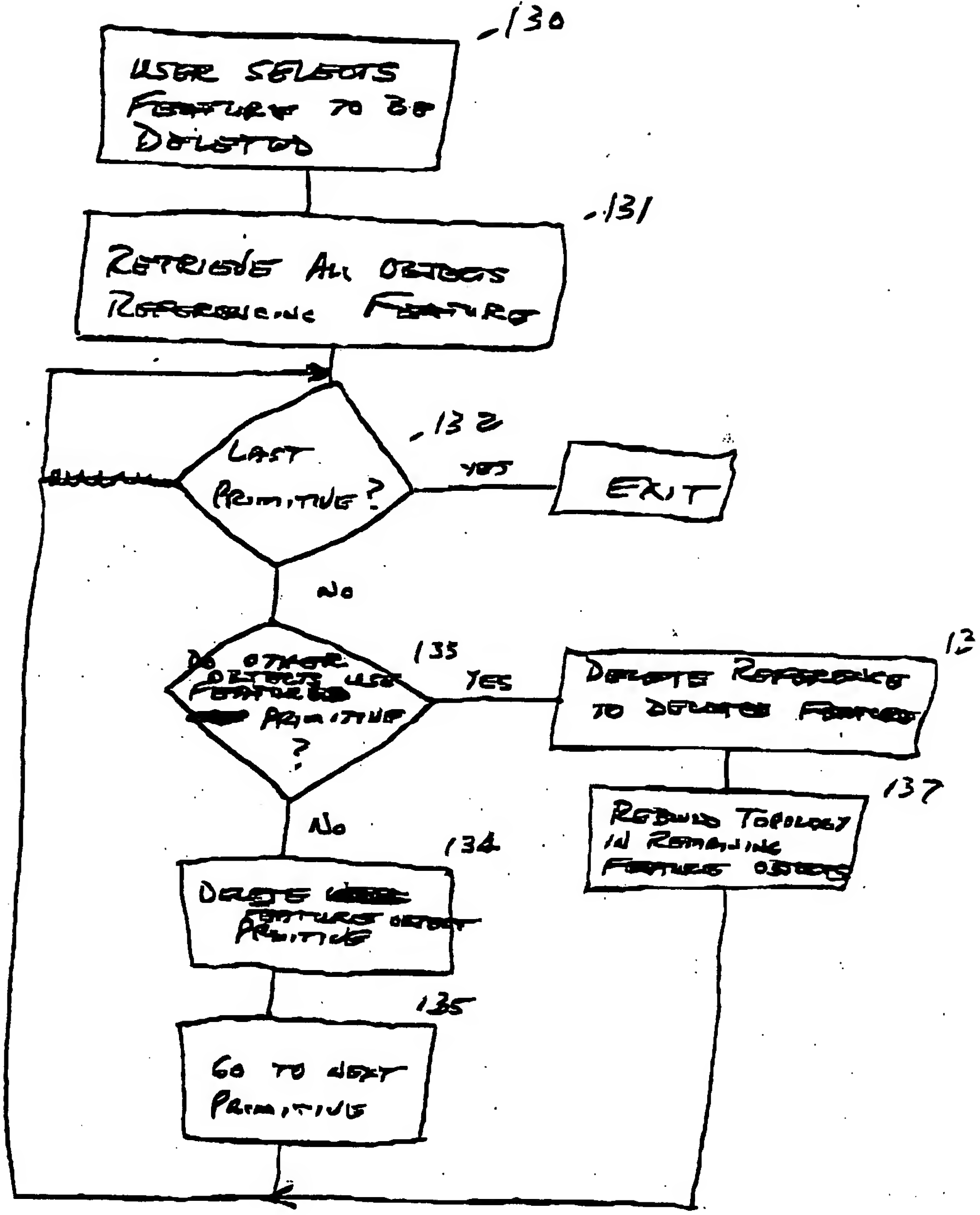

130
USER SELECTS FEATURE TO BE DELETED
131
RETRIEVE ALL OBJECTS REFERENCING FEATURE
132
LAST PRIMITIVE?
YES
EXIT
NO
135
DO OTHER OBJECTS USE FEATURE PRIMITIVE ?
YES
13
DELETE REFERENCE TO DELETED FEATURE
137
REBUILD TOPOLOGY IN REMAINING FEATURE OBJECTS
NO
134
DELETE FEATURE OBJECT PRIMITIVE
135
GO TO NEXT PRIMITIVE

FIG 13